Can ISPs take the heat from Overlay Networks?

Ram Keralapura[1,2], Nina Taft[2], Chen-Nee Chuah[1], Gianluca Iannaccone[2]
[1]Univ. of California at Davis, [2]Intel Research

ABSTRACT

Overlay networks attempt to take control over routing in order to achieve better performance in presence of network failures or congestions. ISPs do the same by employing common traffic engineering techniques such as link weight settings, load balancing and routing policies. In this paper, we examine some of the interaction dynamics between the two layers of control from an ISP's view. With the help of simple examples, we illustrate how an uncoordinated effort of the two layers to recover from failures may cause performance degradation and unpredictable behavior. We also show how current traffic engineering techniques are inadequate to deal with emerging overlay network services.

1. INTRODUCTION

Overlay networks have emerged as a promising platform to provide customizable and reliable services at the application layer to support multicast (e.g., SplitStream [5]), content delivery (e.g., Akamai [1]), resilient connectivity (e.g., RON [4]), and distributed hash table services [13, 15], among others.

Overlay networks typically consists of pre-selected nodes, located in one or multiple network domains, that are connected to one another through application-layer routing. One of the underlying paradigms of overlay networks is to give applications more control over routing decisions, that would otherwise be carried out solely at the IP layer. The advent of a wide variety of active measurement techniques has made this possible. An overlay network typically monitors multiple paths between pairs of nodes and select one based on its own requirements of end-to-end delay, loss rate, or throughput.

Allowing routing control at both the application and the IP layers could have profound implications on how Internet Service Providers (ISPs) design, maintain, and run their networks. The network architecture, along with the set of algorithms and *traffic engineering* (TE) policies that ISPs use, are based on certain assumptions about how their customers and traffic behave. Overlay networks could call into question some of these assumptions, thus rendering it more difficult for ISPs to achieve their goals. For example, it is very important for an ISP to perform load balancing across *all* of its links to ensure good performance and also to limit the impact of failures. We hypothesize that the co-existence of multiple overlays could severely hamper an ISP's ability to achieve this goal.

Another TE issue stems from an ISP's need to estimate its *traffic matrix* (TM). ISPs need to understand the traffic demands for many TE tasks such as capacity planning, reliability analysis, and link weight assignment. We believe that overlay networks could make an ISP's job of estimating its TM more difficult, resulting in erroneous decisions by the ISP.

In addition, routing decisions at two independent layers could lead to short-term or long-term traffic oscillations. If overlay networks react to events in the IP network (e.g., failures or congestion) independently of an ISP, race conditions could occur and lead to traffic oscillations. Such traffic oscillations not only affect the overlay traffic but also impact the background (non-overlay) traffic in the network. We also show that such reactions by overlay networks that span multiple domains could threaten the effectiveness of BGP in isolating different domains in the Internet.

These critical issues raise an important question: *Can overlay networks and underlying IP networks form a synergistic co-existence?* Given the increasing popularity of overlay networks, it is critical to address issues that arise from the interaction between the two layers. We hypothesize that it could be problematic to have routing control in two layers, when each layer is unaware of things happening in the other layer. ISPs may neither be aware of which nodes are participating in an overlay, nor their routing strategy. Overlay networks are not aware of the underlying ISP's topology, load balancing schemes, or timer values for failure reaction.

Qiu et al. [12] describe the interactions between overlay networks and ISPs after the routing control mechanisms reach the Nash equilibrium point. In this paper, we explore some of the issues that arise due to dynamic interactions in the presence of unexpected or unplanned events such as network failures. We believe that the dynamic network behavior in the presence of failures (that are common, everyday events for ISPs [8]) is very important for ISPs and needs to be addressed.

In the rest of the paper, using simple illustrations, we show that these dynamic interactions could be problematic and make the case for future research in this direction. It is important to note this paper is not about performance issues measured by the classic metrics such as loss, delay or throughput. Instead we are interested in understanding how the network management techniques currently deployed by ISPs are impacted by overlay networks. We believe that ISPs need to clearly understand the implications of overlay network behavior, and if needed, develop mechanisms or new services to handle overlay network traffic.

2. MODELING AND SIMULATING ROUTING

To quantify the interactions between the overlay layer and the underlying IP layer, we built a Java-based control plane simulator to analyze (a) the conflicts in decisions made by two different layers and (b) the impact of such decisions on the data traffic.

2.1 Overlay Network Dynamics

While different overlay networks designed for a wide range of applications may differ in their implementation details (e.g., choice of topologies or performance goals), most of them provide the following common set of functionalities: path/performance monitoring, failure detection and restoration. In our simulation model, we attempt to capture the most generic properties of an overlay network:

• The routing strategy in most overlay networks is to select the path between a source and a destination with the best performance based on end-to-end delay, throughput, and/or packet loss. For example, RON and Akamai select their paths based on all the three parameters, while some peer-to-peer applications only consider end-to-end delay. **Our model:** We assume that the overlay network will select the path with the shortest end-to-end delays.

• Most overlay networks monitor the paths that they are using by sending frequent probes to make sure the path adheres to acceptable performance bounds. For instance, RON probes all its paths every 12 seconds while Akamai does the same at a sub-second time scale. **Our model:** In our simulation we consider the probe interval to be x time units (we use generic *time units* because we are more concerned about the ratio and relative values of timers rather than network specific values).

• If the probing event detects a problem with a path (due to various events like failures, congestion, etc. in the IP network), then the overlay network sends probes at a higher rate to confirm the problem before selecting an alternate path. **Our model:** If a probe does not receive a response within a given *timeout* value, then the overlay network queries the path at a higher rate (every y time units) to ensure that the path is bad. If a path remains bad after n such high frequency probes, the overlay network will then find an alternate path (in this case, the next best path) between the source and destination nodes. As soon as an alternate path is found, the traffic is moved to the alternate path, which is now probed every x time units to ensure that it is healthy.

Note that all the parameters, x, y, n, and *timeout*, are configurable and can be set to different values to simulate the routing strategies of different overlay networks. In our simulations, multiple overlay networks can be simulated, each with its own topology and routing strategy.

2.2 IP-Layer Routing Dynamics

Within each domain, we emulate an IP-layer *interior gateway protocol* (IGP) that implements Dijkstra's shortest path algorithm. We generate link failures and model IGP dynamics in response to failures as outlined in [8] and [9].

In any IP network, the link utilization level determines the delay, throughput, and losses experienced by traffic flows traversing the link. To simulate realistic link delays, we use a monotonically increasing piecewise linear convex function similar to [7] and [12]. Note that in all our test scenarios, we assume that a load of 50 units on a link is the threshold value beyond which it exhibits very high delay values.

In the following sections, we identify numerous issues that result in potentially harmful interactions between the IP networks and the overlay networks, and explore each of them in detail to make the case for further research in this area.

3. CHALLENGES TO TE

ISPs deploy TE techniques to control how the traffic flows in the network and balance load across the network. In general, overlay networks attempt to bypass ISP-dictated path and find alternate paths that maximize their own performance. Qiu et al. [12] have shown how *selfish* routing in overlay networks may have a negative impact on intra-domain TE, i.e., overlay networks tend to overload links on the shortest paths.

ISPs apply TE mainly in reaction to changes in the topology (due to link failures [7, 11]) or traffic demands (due to flash crowd events or BGP failures). In these scenarios, a common way for ISPs to manage the traffic is by changing the IGP link weights. ISPs make two assumptions while using this technique: (i) traffic demands do not vary significantly over short timescales, and (ii) changes in the path within a domain do not impact traffic demands. Overlay network routing defeats these assumptions as illustrated below.

3.1 Traffic Matrix Estimation

ISPs employ techniques such as [14] to compute their TM i.e., a matrix that specifies the traffic demand from origin nodes to destination nodes in a network. A TM is a critical input for many traffic engineering tasks, such as capacity planning, failure analysis, link weight setting, etc.

Conventionally, overlay nodes typically encrypt the information about the final destination in the data packets and hence the IP layer is unaware of the true destination. Traffic between two overlay nodes could traverse multiple overlay hops. At each hop, traffic exits the IP network and reaches the overlay node, which deciphers the next overlay hop information and inserts the traffic back into the IP network. Consider the network in Figure 1(a), which has multiple overlay nodes (at A, B, D and G) in a ISP domain. Suppose the traffic between nodes A and D is 10 units. If IP routing is used then the TM entry for the source-destination pair $A-D$ is 10. But if overlay routing intervenes and decides to route the traffic through an overlay path (say, $A-B-D$) which offers better latency, then the TM entry for the pair $A-D$ is *duplicated* as two entries of 10 units each, one for $A-B$ and another for $B-D$, while the value for the entry $A-D$ is 0. This implies that overlay networks could often change TM values at short time scales (i.e. introduce dynamism), thus requiring the ISP to perform frequent estimation of TM to maintain its accuracy.

There are many flows whose ultimate destination lies outside the ISPs' domain; the traffic from these flows traverses the ISP and thus appears inside the TM. The traffic matrix will specify an exit router within the ISP's domain for such

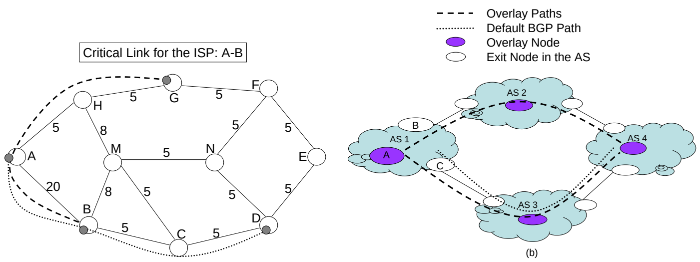

Figure 1: Illustration Networks

Scenario-1				
Timer	x	y	n	*timeout*
Overlay-1	310	150	3	100
Overlay-2	300	150	3	120
Scenario-2				
Timer	x	y	n	*timeout*
Overlay-1	500	150	3	100
Overlay-2	500	150	3	110
Overlay-3	200	150	3	100

Table 1: Overlay Timers for *Scenarios 1 and 2*

flows. If this traffic belongs to an overlay network that spans multiple domains, and uses its own path selection mechanism, then the exit point within a single ISP domain could change, resulting in a *shift* in TM entry. For example, consider the network in Figure 1(b). Suppose that the layer-3 path from $AS1$ to $AS4$ is through $AS3$. If the overlay network discovers a better path through $AS2$, then the overlay network could switch the routing path, thus changing the associated exit point. The TM entry in $AS1$ for $A - C$ now *shifts* to $A - B$. If this were to happen for large flows, it could affect a significant portion of the TM. If this were to happen often, it would increase the dynamic nature of the TM. TM entry *duplication* and *shift* are two examples that warrant frequent TM updates.

3.2 Load Balancing

Another important traffic engineering task is to define intra-domain routing policies. For example, considering again the network in Figure 1(a). Suppose that the ISP has two important customers connected to nodes A and B, and hence assigns a high link metric to link $A-B$ (as shown in Figure 1) to discourage the use of this link by other node pairs. Suppose that node D wants to send traffic to node A. Using IP-layer routing, the path traversed by the traffic would be $D-N-M-H-A$. However the overlay node D can choose to reach A by forwarding through another overlay node, such as B. In this case, the path $D-C-B-A$ is used. Similar choices can be made for the traffic from A to D, B to G and G to B. This undermines the ISPs intent, and an ISP could thus erroneously assume that the majority of resources on link $A - B$ are used by its two important customers. The impact of bypassing routing or load balancing policies could be magnified when multiple overlay networks co-exist and make independent decisions.

4. RACE CONDITIONS IN MULTIPLE OVERLAY NETWORKS

Overlay networks attempt to provide "enhanced" services to applications by routing their traffic through paths that adhere to strict performance constraints. A degradation in path performance will trigger overlay networks to find an alternate path that satisfies the performance constraints and re-route the traffic accordingly. If multiple overlays co-exist then a performance degradation event will trigger a reaction in all overlays traversing the same problematic spot in the network. Two or more overlays reacting at moments that are close in time can result in race conditions. Having routing control in two layers in the network is equivalent to having two closed loop systems reacting simultaneously yet independently to the same set of events. This is a classic situation for race conditions that lead to traffic oscillations.

We will see that there are a number of events that trigger oscillations, such as link or node failures, IGP convergence, and congestion (high loads). There are also a number of ways and/or events that cause oscillations to stop, such as self-disentangling (explained below) and failure restoration. The different combinations of such start and stop triggers means that there exists a variety of scenarios in which oscillations occur and that oscillations can last for varying amounts of time, some short and some long.

The co-existence of multiple overlays has not been well explored. We believe that it increases the likelihood of harmful traffic oscillations that could affect a significant portion of traffic due to race conditions. We consider three simple examples to explore these ideas on race conditions and multiple overlays. Although our examples consider small test topologies to identify and illustrate different possible interactions, our observations would apply to large networks when a sub-graph of the network resembles our test topologies.

We first consider a scenario with two overlay networks on top of a single ISP domain (*Scenario 1*, Figure 2(a)). The IP network consists of 7 nodes and 8 links while the two overlay networks on top of the IP network have 4 nodes each with mesh connectivity. The numbers on the links in Figure 2 represent the link loads. While the numbers enclosed in the boxes and circles represent the overlay traffic load, the non-enclosed numbers represent the background traffic in the IP network. We assume that the only traffic in both the overlay networks is from node A to node D and is equal to 20 units each. The timer values (as in Section 2) used for the overlay networks are shown in Table 1. Note that these values are similar, but not the same, for both overlay networks. For example, two overlays that both cater to video streaming could end up with similar application level timer values.

Initially the overlay traffic between nodes A and D traverses the IP path $A - C - D$ (Figure 2(a)). Notice that the link loads on all the IP links are below the congestion threshold value of 50 units. Now consider the event that link $A - C$ experiences a failure. If both overlay networks react faster than the IP layer, they might independently decide to move their traffic to the top path. This can happen if previous probing indicated that the path $A-B-H-D$ offers better performance than the bottom path $A-E-F-D$. In other words, the first overlay network decides to reroute

its traffic through B and the second overlay network decides to reroute its traffic through H without being aware of each other's traffic shift.

This in turn results in very heavy load on links $A-B$ and $H-D$. Both the overlay networks react to this congestion and find a new path to reach the destination node D from A. Again both the overlay networks decide to move the traffic to the bottom path at nearly the same time, as shown in Figure 2(c). These traffic shifts create overload on links $A-E$ and $F-D$. Once again both overlay networks react to this by re-routing the traffic back to the top path. This results in traffic oscillations between the top and bottom paths until one of the overlay networks reacts and reroutes traffic faster than the other (Figure 2(d)), thus breaking the deadlock.

The resulting oscillations for *Scenario 1* are depicted in Figure 3 that shows the utilization of various links in the IP network across time. We see that soon after the link failure event, loads on links $A-B$, $A-E$, $B-H$, $E-F$, $H-D$ and $F-D$ start oscillating. In this case the oscillations stop before the IGP protocol converges. Note that the same reaction could have been triggered by a flash crowd event rather than a failure. A traffic surge targeting a server at node D can create sufficient congestion to seriously degrade the path quality between $A-D$ and thus engender the same reaction.

This type of scenario can happen when the path probes from the two overlays end up being spaced close in time; in essence the two overlay networks can get *synchronized* in their detection of "better alternate paths" and in their traffic shifting. In this case the traffic shift in one overlay network is not visible to the high frequency probes of the other. Even though probes in different overlays are initiated at different points in time, it is possible for them to get synchronized. If the inter-probe times are similar, but not exactly the same, then over time the alignment between two such probes will grow, separating them out enough to break the synchronization. At this point, one overlay will detect congestion and move its traffic before the second one does. If the second overlay detects the drop in load (since the first one moved) via its high frequency probes, then it no longer moves. When this happens we say that the two overlays *disentangle* themselves. Our observation that overlays can become synchronized reveals a behavior similar to that presented in [6] on general synchronization of periodic routing messages. From that work we surmise that the cause of synchronization may lie in the periodic nature of the overlay probing processes.

We now consider *Scenario 2*, with the same IP network, but add one more overlay network as shown in Figure 4(a). Similar to *Scenario 1*, the only overlay traffic in all the three networks is between nodes A and D. Table 1 shows the timer values for the three overlay networks. The third overlay network probes the network more often than the other two, thus reacting faster to performance degradation events.

Figure 4(b) shows the network state after the link $A-C$

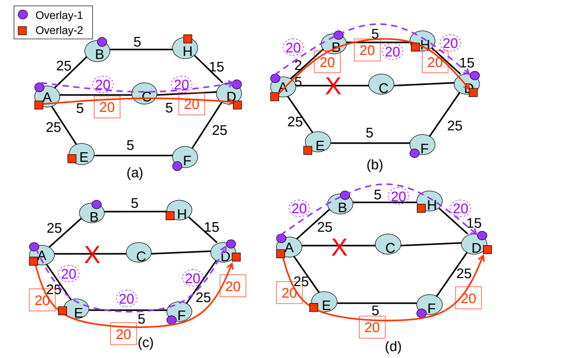

Figure 2: Scenario 1: Two overlays in one domain

fails. Note that the third overlay network reacts first and chooses the top path while the other two networks choose the bottom path. At this point in time, none of the links are overloaded and all the overlay networks have found a stable path. However, after the IGP protocol converges, the new layer-3 path selected from node A to node D is $A-B-H-D$. The first two overlay networks are oblivious to the fact that the underlying path between the source and destination nodes has changed and hence assume that the original overlay link $A-D$ has recovered. The networks start probing the overlay link $A-D$ (i.e. the IP path $A-B-H-D$) and find that it offers better performance than their current path (i.e. $A-E-F-D$) due to the fact that the links along the path $A-B-H-D$ are less loaded than $A-E-F-D$. The first overlay network reacts faster than the second overlay network and moves to the new path. This results in a situation as shown in Figure 4(d), where the first and third overlay networks use the top path and the second overlay network uses the bottom path. This overloads the link $A-B$ and the third overlay network reacts to this link overload faster than the first overlay network, resulting in the situation depicted in Figure 4(c). This traffic shift overloads link $A-E$, leading to oscillations. The oscillations stop when the overlay networks land in the situation where the first two overlay networks use the same path and the third one uses the other path (Figure 4(b)).

Our simulation tracks the dynamic evolution of such a scenario. Figure 5 shows the utilization of various links during the oscillations. We see that the network is stable until IGP re-converges, after which loads on several links

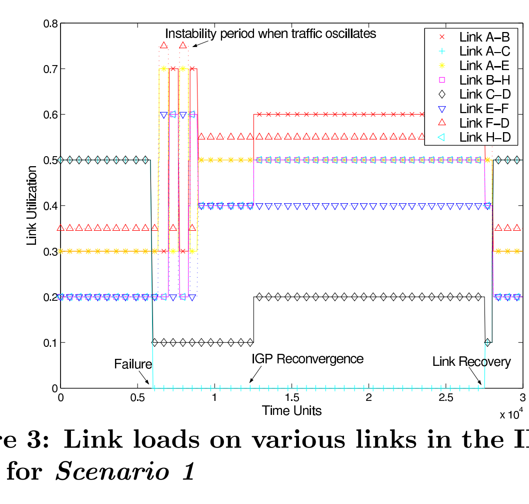

Figure 3: Link loads on various links in the IP network for *Scenario 1*

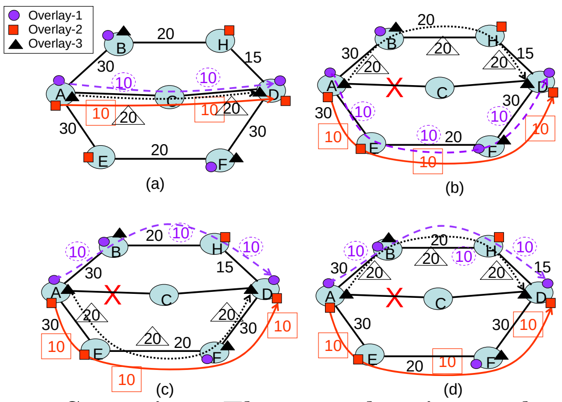

Figure 4: Scenario 2: Three overlays in one domain

start oscillating and continue until the overlay networks disentangle themselves. Note that in this scenario the trigger for oscillations was the IGP convergence event, whereas in the previous scenario the trigger was the failure itself.

Whether or not oscillations happen, and how long they last, depend very much upon the arrival times and temporal inter-spacing of the probes at the failure event. We considered a variation of *Scenario 2* in which we altered the random start times of each of the overlay probes, and changed the link loads slightly. Due to the way the probes ended up being aligned, oscillations occurred as depicted in Figure 4(c). Soon after the link failure, loads on many links start oscillating, as shown in Figure 6. These oscillations continue even after the IGP re-convergence event, and stop only when the overlay networks disentangle themselves and reach a state similar to the one shown in Figure 4(b).

From these simple examples, we conclude: (1) Traffic oscillations involving multiple overlay networks can be short-term or long-term depending on the network state, overlay timer values and their synchronization at the time of the event that triggers oscillations. (2) A variety of events occurring at the IP layer like IGP re-convergence, failure recovery, self-disentangling, etc., can influence the start or stop of oscillations at the overlay layer. Due to lack of space, we only show the disentangling examples here. However, we have used our simulator to study other scenarios where oscillations do not stop until either IGP convergence or failure recovery. We remind the reader that these observations are based on one case in which the probe rates of two overlays are very similar, and a second case in which

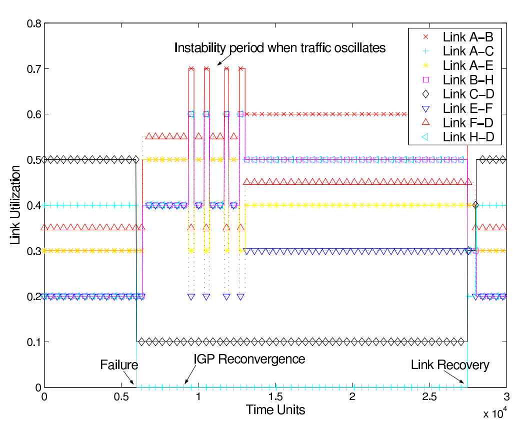

Figure 5: Link loads on various links in the IP network for *Scenario 2*

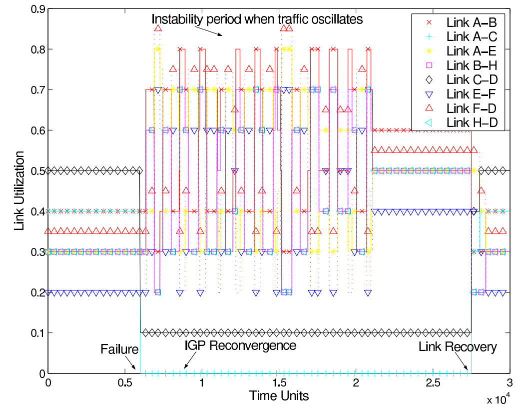

Figure 6: Link loads on various links in the IP network for *Scenario 2*

the ratio of the probe rates among overlays is 2:1.

Overlay networks route traffic independently of the underlying IP network, ignoring the potential impact on the background traffic. Figure 7 shows the end-to-end delay experienced by the traffic from A to H in *Scenario 2*. The background traffic experiences highly variable end-to-end delay that could result in jitter, service disruption and packet losses. This is a serious problem for ISPs since they are held responsible for the performance of *all* their customer traffic, including non-overlay traffic.

5. COUPLING OF MULTIPLE AS DOMAINS

We next consider a scenario with a single overlay network that spans multiple domains (*Scenario 3*, Figure 8(a)). The overlay path between the nodes A and G is $A-F-G$ and the underlying path is $A-B-F-G$. Now consider the event that link $F-G$ fails in 'Domain-2'. Figure 8(b) shows the state of the network soon after this failure. The new overlay path between A and G is $A-C-G$. This results in overloading the link $A-C$. The overlay network reacts to this overload and finds another alternate overlay path through H (Figure 8(c)). This traffic shift overloads the link $H-G$ and hence the overlay network moves the traffic back to the overlay path $A-C-D$. This results in traffic oscillations in 'Domain-1' (and 'Domain-2') until the IGP converges in 'Domain-2'. At this point, the overlay network finds the stable overlay path $A-F-G$ and the oscillations stop (Figure 8(d)).

From this scenario we can infer: (1) When an overlay

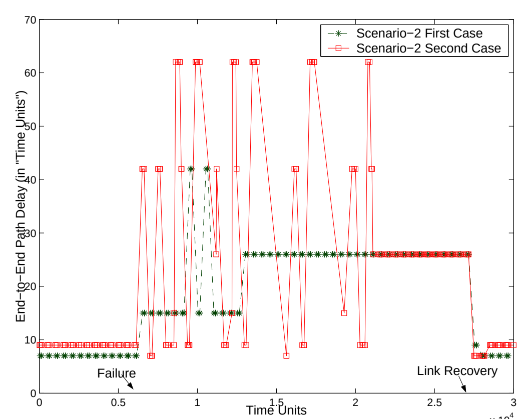

Figure 7: End-to-End Delay for Path A-H in *Scenario 2*

network spans multiple domains, the network state in one domain could influence the network behavior in another domain. (2) One of the aims of BGP is to decouple different domains so that events in one domain do not affect another domain. Overlay networks can defeat this objective by inducing a coupling of the two adjacent ASes. An event in one domain may result in traffic oscillations in the other.

6. DISCUSSION

We have identified five problematic interactions that can occur between IP networks and overlay networks: (*i*) traffic matrices become more dynamic and more ambiguous, making them harder to estimate; (*ii*) some types of load balancing policies can be bypassed; (*iii*) multiple overlays can get synchronized, which in turn leads to traffic oscillations that can last for varying amounts of time; (*iv*) oscillations can impact non-overlay traffic; and (*v*) different ASes can get coupled due to per-domain events.

We believe that it is imperative for ISPs to have better knowledge about overlay networks to cope with the interactions and provide good service to *all* its customers. The scenarios raised here imply that traditional traffic engineering techniques may not be sufficient for this purpose when overlay networks become widespread. In order for ISPs and overlays to form a more synergistic co-existence, ISPs may want to think about how to design incentives for overlay applications to avoid behaviors that are problematic. Similarly it is important for overlay networks to understand the implications of their routing strategies and adopt mechanisms to curb harmful interactions. For example, adding a random component to probe timeouts could minimize the likelihood of synchronization and hence load thrashing.

When the original Internet was developed, it was designed as an overlay on top of X.25 networks. Although we may try to draw some lessons learned by revisiting that history, we suspect the analogy may be thin for two reasons. First, the Internet was designed as a single overlay, whereas now we are looking at a situation with potentially many overlays co-existing. Secondly, the goal of the Arpanet was to provide an additional service (i.e., connectionless best-effort communications) that the underlying networks were not providing. The routing control and failure restoration in overlays are *competing* with the same kind of services offered at the IP layer. We could also draw

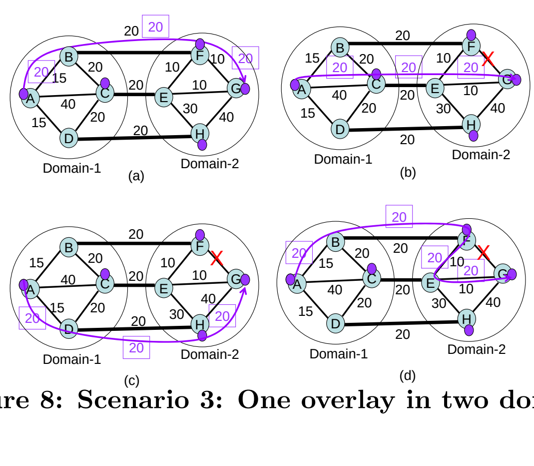

Figure 8: Scenario 3: One overlay in two domains

upon lessons learned from the design of IP over SONET or DWDM networks [3, 8] in which the division of labor is carefully thought out such that each layer is responsible for different kinds of failures.

We thus believe that the types of issues raised herein should be addressed within the research community sooner rather than later. Some interesting topics for future research in the area of integration between ISPs and overlay networks include:

- Evaluate the benefit of using an overlay service compared to augmenting IP-layers with other route control techniques, such as multi-homing as in [2].
- Develop a formal framework for understanding overlay network behavior that would help distinguish general problems versus pathological cases.
- Define mechanisms to allow different overlay networks to share routing and performance information among themselves. Such coordination among networks (similarly to how BGP allows coordination between ISPs) could help avoid synchronization events that lead to traffic oscillations. One step in this direction has been presented in [10] where a routing "underlay" is proposed to enable sharing of performance measurements between different overlays. The only risk with a routing underlay is that it could increase the likelihood of synchronization between overlay networks. Introducing randomness in the overlay network timers or routing underlay responses could mitigate this situation.
- Define mechanisms to share information between the overlay and IP networks so that the two networks can be aware of the events and state information affecting each other.

7. REFERENCES

[1] Akamai. http://www.akamai.com.

[2] A. Akella, J. Pang, B. Maggs, S. Seshan, and A. Shaikh. A Comparison of Overlay Routing and Multihoming Route Control. In *Proceedings of ACM Sigcomm*, Sept. 2004.

[3] American National Standard for Telecommunications. Technical report on enhanced network survivability performance. ANSI T1.TR.68, 2001.

[4] D. Anderson, H. Balakrishna, M. Kaashoek, and R. Morris. Resilient Overlay Networks. In *SOSP*, Oct. 2001.

[5] M. Castro, P. Druschel, A. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. SplitStream: High-Bandwidth Multicast in Cooperative Environments. In *SOSP*, Oct. 2003.

[6] S. Floyd and V. Jacobson. The Synchronization of Periodic Routing Messages. *IEEE/ACM Transactions on Networking*, Apr. 1994.

[7] B. Fortz and M. Thorup. Internet Traffic Engineering by Optimizing OSPF Weights. In *Proceedings of IEEE Infocom*, Mar. 2000.

[8] G. Iannaccone, C.-N. Chuah, S. Bhattacharyya, and C. Diot. Feasibility of IP Restoration in a Tier-1 Backbone. *IEEE Network*, Mar. 2004.

[9] R. Keralapura, C. Chuah, G. Iannaccone, and S. Bhattacharyya. Service Availability: A New Approach to Characterize IP Backbone Topologies. In *IEEE IWQoS*, June 2004.

[10] A. Nakao, L. Peterson, and A. Bavier. A Routing Underlay for Overlay Networks. In *ACM SIGCOMM*, Aug. 2003.

[11] A. Nucci, B. Schroeder, S. Bhattacharyya, N. Taft, and C. Diot. IGP Link Weight Assignment for Transient Link Failures. In *ITC*, 2003.

[12] L. Qiu, Y. Yang, Y. Zhang, and S. Shenker. On Selfish Routing in Internet-Like Environments. In *ACM SIGCOMM*, 2003.

[13] S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz. Handling Churn in a DHT. In *USENIX Annual Technical Conference*, June 2004.

[14] A. Soule, A. Nucci, E. Leonardi, R. Cruz, and N. Taft. How to Identify and Estimate the Largest Traffic Matrix Elements in a Dynamic Environment. In *ACM SIGMETRICS*, June 2004.

[15] B. Zhao, L. Huang, J. Stribling, A. Joseph, and J. Kubiatowicz. Exploiting Routing Redundancy via Structured Peer-to-Peer Overlays. In *ICNP*, 2003.